



TITLE:

自然な数学記述によるグラフ作成 ツールの提案 (数学ソフトウェアと その効果的教育利用に関する研究)

AUTHOR(S):

富永, 侑里; 遠藤, 菜津美; 福井, 哲夫

CITATION:

富永, 侑里 ...[et al]. 自然な数学記述によるグラフ作成ツールの提案 (数学ソフトウェアとその効果的教育利用に関する研究). 数理解析研究所講究録 2019, 2105: 69-78

ISSUE DATE:

2019-02

URL:

<http://hdl.handle.net/2433/251875>

RIGHT:

自然な数学記述によるグラフ作成ツールの提案

武庫川女子大学・生活環境学部 富永 侑里, 遠藤 菜津美, 福井 哲夫
Yuri Tominaga, Natsumi Endo, Tetsuo Fukui,
School of Human Environmental Sciences, Mukogawa Women's University

1 はじめに

デジタル教科書が, 2019年4月から小中高校などで使えるようになり, 文部科学省は, 小学校で次期学習指導要領が全面実施される2020年度から本格的に学校現場での活用を促す考えである. デジタル教科書の導入メリットとして文部科学省は, 基本的なデジタル機能の活用による教育活動の充実と, デジタル教材との一体的使用が可能になることの2点を挙げている [1]. 特に, 動画・アニメーション教材やデジタル端末上でドリルワークが行える教材といったデジタル教材を, デジタル教科書と合わせて使うことで内容をより理解しやすくなることから, 我々はデジタル教材の重要性に着目した. また, 数学は他の教科に比べて, 考えて解くドリルワークが必要であることから, 解答過程やその確認に使用できるようなデジタル教材として「生徒・学生がデジタル端末上で手軽にグラフを作成できるツール」が重要な役割を果たすと考え.

しかし既存のツールでは, 関数のグラフを定義する数式の入力はシステムの決めた文法で操作する必要がある, 数式表示も1次元の文字列表記のままであることも多い. そのため, 数学的に考えるためのツールとしては, 初学者にとって不自然で, 操作の負担が大きい.

著者の一人である福井は, 数式の取扱いを身近にするために, 数式を読むような曖昧な記述からシステムが最適候補を予測して入力できるインタフェース MathTOUCH を2012年に提案した [2, 3].

本研究では, 数学教材作りや数学学習の道具としてのグラフ作成ツールにおいて, 関数グラフなどを作成するために必要な数式入力の負担を改善するため MathTOUCH を実装することを提案し, 自然な数学記述によるグラフ作成ツールを試作したので, 報告する.

2 従来のグラフツールと問題点

既存の数学的なグラフ作成ツールには様々なジャンルのものがある. MacOS が提供する Grapher[4] のように関数グラフを描くために専用で作られているものや, 数学教育の分野ではよく使用される GeoGebra[5] などの動的幾何作図ツールもその一つである. また, Mathematica[6] のような数式処理システムにも本格的なグラフ描画機能が備わっている.

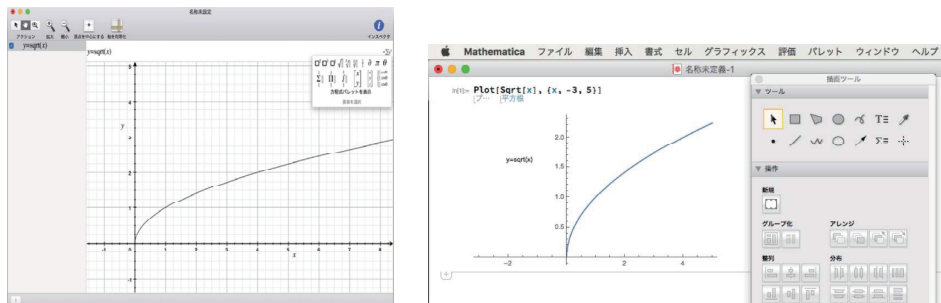


図 1: 関数グラフ専用ツール Grapher(左) と Mathematica によるグラフ描画 (右)

しかし、例えば図1(左)のように関数 $y = \sqrt{x}$ を入力する場合、Grapherでは「 $y=\text{sqrt}(x)$ 」という関数を入れるか、右上のパレットから「 $\sqrt{\quad}$ 」記号を選ぶといった作業が必要になる。しかしこれでは、グラフを作成するために関数とは別のコマンドを覚えることになる。また、パレットから選ぶという操作も複雑な式の場合、面倒であるため、初学者には負担になると考えられる。

一方 Mathematica は、コマンドで様々な代数計算やグラフの描画などが可能で、描画したグラフもきれいで高性能である。しかし、Grapherと同様に、数式などの入力がプログラミング的で読みにくいこと、グラフの描画命令文がコマンドであることから、描きたいグラフの方程式が複雑なほどプログラムによる数学とは別の視点を考えなければならなくなることが問題である。

また、図1(右)の右側のようなパレットから点や矢印などの図形オブジェクトをグラフ内に挿入できるが、これらはすべてマウスで移動させるので、座標は近似値になってしまう。そのため、例えば点 $(\sqrt{3}, \sin \frac{\pi}{3})$ のような場合では正確な数学的座標指定が難しい。同じくパレットからテキストオブジェクトも挿入できるので、複数のグラフを描いたときなど、グラフの方程式をテキストで挿入することは可能であるが、「 $y=x^2$ 」のように、表示が2次元的(実際は $y = x^2$)でない。

以上のように、従来のグラフツールの問題点を整理すると、以下の5つにまとめられる。

問題点 (1) 数式入力がプログラミング的で、特殊な文法に従うことが多い。

問題点 (2) 2次元的な数式を数学的表記のままグラフ内に表示するのが難しい。

問題点 (3) 点やベクトルなどの図形オブジェクトの座標が近似値しかできない。

問題点 (4) 図形描画命令文が、数学的に(どんなグラフや図形を描いているか)意味を把握しにくい。

問題点 (5) 図形描画命令文が英語的で、日本語で命令できる形になっていない。

3 グラフ作成ツールの設計方針と解決策

ここでは第2章で述べた問題点(1)から問題点(5)を解決するようなグラフ作成ツールを開発するための設計方針について議論する．その中で特に重要な役割を果たす，数式入力インターフェース MathTOUCH については3.2節で解説する．

3.1 グラフ作成ツールの設計方針

まず，目的とするグラフ作成ツールは，教師のPCや教室のタブレット端末で動作し，ツール導入時に技術的知識がなくても良いように Web アプリケーション (HTML5) として作成したい．そのため，作図・画像処理に canvas[7] を使う．

問題点(1)の数式入力の負担軽減については MathTOUCH の研究がある [8, 9]．MathTOUCH は数式の入力を読むような曖昧な表現で入力すると，AI が候補を予測し変換する数式入力インターフェースである．AI が候補を予測するため，コマンドを覚えることやパレットから選ぶ手間がなく負担が少ない [10]．よってこれを利用する．MathTOUCH の詳細な利用方法については次節で述べる．

問題点(2)については，MathTOUCH で構成された数式は木構造による内部表現で保管され， \LaTeX や PNG などの様々な数式に変換可能で，Web ブラウザ上の canvas にも教科書どおりの2次元表記の数式を描画できる．

問題点(3)，(4)，(5)を考慮して，本ツールでは関数グラフ，座標軸，点やベクトルなど図形要素をそれぞれ1行のスク립トで記述してもらい，各行の文をシステムが解釈することで図形を描画する方式を採用する．

問題点(3)の原因は，従来のツールのほとんどが点やベクトルなどの図形要素を手軽に配置できるようにマウス操作で行うことにある．本研究では，点やベクトルも重要な数学要素と考え，それらの座標も厳密な数式，例えば点 $(\sqrt{3}, \sin \frac{\pi}{3})$ のように記述する方式を採用する．本来，点を描くために座標を数式で入力するのは面倒であるが，MathTOUCH の採用によって負担が軽減されれば，逆に強力なツールとなる．

また，図形要素スク립トの文法は，例えば「方程式 $y = \frac{1}{3}x^2$ のグラフ」のように，日本語の数学教科書に則した表現で入力してもらい，特に数式部分も教科書どおりの2次元表記の数式で表すこととする．図形要素スク립トの日本語入力は，HTML の `<div>` 要素の `contenteditable` 属性 [7] によって編集可能であり，数式の挿入には MathTOUCH を介して PNG 形式の `` 要素として実現する．

3.2 数式入力インターフェース MathTOUCH

ここでは数式入力部で使用した MathTOUCH [2, 3, 8, 9, 10] について解説する．MathTOUCH は著者の一人である福井が開発した数式入力インターフェースである．JavaScript で開発され，Web ブラウザ上で動作する Web アプリケーションとして利用できる．数式の入力を，キーボードにより仮名漢字変換を行うように変換候補を選択しながら入力する．数式は一般に2次元構造をもち，1次元的な文字列だけでは扱うことができない．

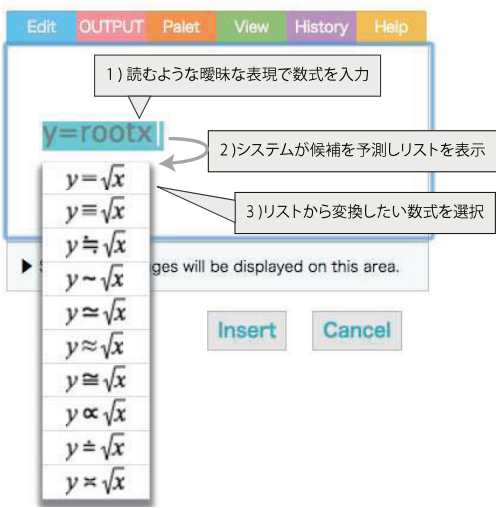


図 2: MathTOUCH による数式の変換

MathTOUCH では図 2 のように、1) 数式を線形文字列で入力すると、2) システムが候補を予測しリストで表示する、3) リストから所望の数式を選択することで 2 次元表記の数式が得られる。図 2 の例では線形文字列「y=rootx」から「 $y = \sqrt{x}$ 」を入力している。

4 グラフ作成ツール

ここでは、本研究で開発したグラフ作成ツールの概要について 4.1 節で述べ、グラフ作成ツールによって作図するための基本機能を 4.2 節で、グラフ作成ツールのプログラム構成とデータ構造およびアルゴリズムについては 4.3 節で解説する。また、本ツールを提案する意義を例示するため、4.4 節で教育利用モデル事例を紹介する。

4.1 グラフ作成ツールの概要

グラフツールの実行画面は図 3 のようになっている。図 3 の例では、「xy 平面」、「座標軸」、「方程式のグラフ」、「数式ラベル」と「点」の 5 つの図形要素が記述されている。

図 3-(1) の図形要素サンプルリストはプルダウンメニューになっており、図 3-(2) の図形要素スクリプト記述エリアがアクティブ (カーソルが表示された状態) のとき、カーソルポイントに選択した「図形要素スクリプトサンプル」が挿入され、同時にその図形要素が図 3-(3) のグラフ表示キャンバスに描かれる。

例えば、方程式を入力したい場合はプルダウンメニューから「方程式のグラフ」を、点は「点」、ラベルは「数式ラベル」を選択することで入力できる。記述する図形要素スクリプトは「方程式 $y = \sqrt{x}$ のグラフ」のように自然な記述で入力できるようになっ

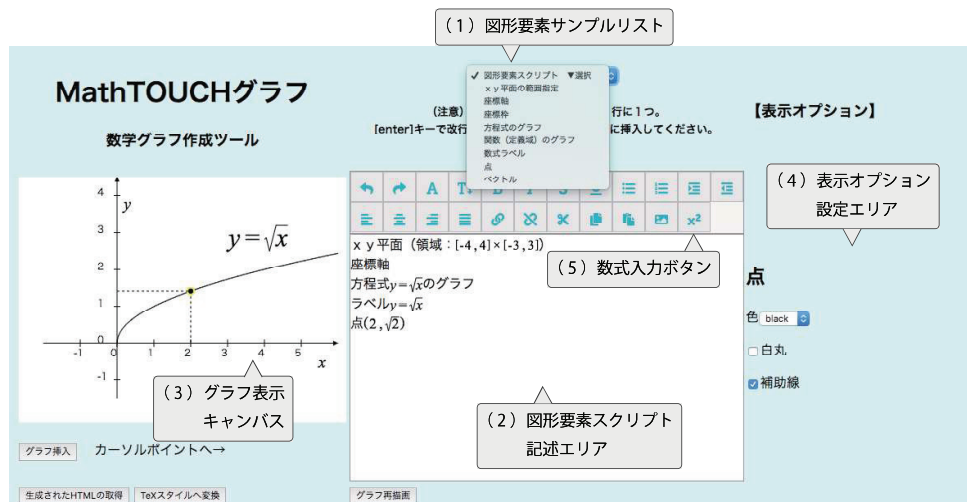


図 3: グラフ作成ツールの実行画面

ている．グラフや点座標のための数式は，図 3-(5) 数式入力ボタン x^2 もしくは '@' の入力で MathTOUCH ウィンドウ (図 2) から入力することが可能である．また，数式はコピー＆ペーストで挿入/置換することも可能である．図形要素スクリプトの順番は，どの図形要素も xy 平面上に描く必要があるため，必ず先に xy 平面の表示範囲を定義し，その後，その他の図形要素を記述する．記述した図形要素は上から順に描画され，複数表示することができる．

図 3-(4) の表示オプション設定エリアでは，線の太さや色など図形要素に応じた表示オプションパラメータを変更することができる．図 3-(3) のグラフ表示キャンバスにある点やグラフの線を選択すると，それぞれに応じた表示オプションが表示される．

4.2 基本機能

ここでは，グラフ作成ツールで利用できるすべての基本機能について解説する．

2018 年 8 月時点の図形要素サンプルリストから選択できる命令は，「 xy 平面の表示範囲」，「座標軸」，「座標棒」，「方程式のグラフ」，「関数 (定義域) のグラフ」，「数式ラベル」，「点」，「ベクトル」の 8 つとなっている．図形要素の指定式，内容を下の表 1 にまとめる．表 1 の 2 列目が図形要素を指示するためのデフォルトのスクリプト文であり，これらは利用者が読みやすいように，本研究の提案である教科書に則した自然な表現を採用している．表 1 の 3 列目の指定式は，図形要素で指定すべき式の形式を表している．グラフの方程式のように，所望する数式を数式入力ボタン x^2 から MathTOUCH (図 2) を介して入力する．

唯一，「 xy 平面の表示範囲」は図形ではなく，表示領域 (矩形区間) を定義するスクリプトであり，必ず他の図形要素より先頭の行に記述しなければならない． xy 平面に続く

表 1: グラフ作成ツールで使用できる基本機能

図形要素	デフォルトのスク립ト文	指定式	内容
xy 平面の表示範囲	xy 平面 (領域: [-4,4] × [-3,3])	[x の区間] × [y の区間]	表示範囲を矩形区間で定義
座標軸	座標軸	なし	x 軸, y 軸の表示
座標枠	座標枠	なし	外枠の表示, 目盛罫線
方程式のグラフ	方程式 $y = \frac{1}{3}x^2$	グラフの方程式 $y = f(x)$	一価関数グラフを描画
関数 (定義域) のグラフ	関数 $y = \frac{1}{3}x^2$ ($0 \leq x \leq 4$)	$y = f(x)$ と (定義域)	定義域の一価関数グラフを描画
数式ラベル	ラベル $y = f(x)$	任意数式	任意の数式を挿入
点	点 (1,2)	(点座標)	指定座標の点を描画
ベクトル	ベクトル (1,2)-(2,-1)	(終点座標)-(始点座標)	指定ベクトルの矢印描画

表 2: 図形要素の表示オプション

図形要素	表示オプション調整項目
xy 平面の表示範囲	領域の変更, 原点移動, 背景色, マウスにより位置・サイズを調整可能
座標軸	各軸・目盛・詳細目盛・軸ラベルの有無, 色, 線幅
座標枠	L 字型, 詳細目盛の有無, 補助目盛罫線 (実線・淡線・点線)
方程式のグラフ	色, 線幅
関数 (定義域) のグラフ	色, 線幅
数式ラベル	マウスにより位置・フォントサイズを調整可能
点	色, 白丸 (塗りつぶしの有無), 補助線
ベクトル	色, 線幅, 線種

スク립ト行に記述された他の図形要素は, (新たに xy 平面が定義されるまで) すべての xy 平面の領域に依存する. 関数グラフは方程式と定義域を別々に修正可能である. ベクトルは (終点座標)-(始点座標) で指定しており, 始点が原点の場合は (終点座標) のみで指定可能である.

また, 図形要素はオプションがあり, これはグラフ表示キャンバス上でオプションを変更したい要素をクリックして選択することで表 2 のようなオプションを変更することが可能である. 例えば, 方程式のグラフでは色や線の太さをプルダウンメニューから変更可能で, 点ではグラフと同様にプルダウンメニューから色を変更したり, 白丸や補助線という項目にチェックを入れることで, 塗りつぶしなしの点にしたり補助線を表示させたりすることが可能である.

4.3 データ構造とアルゴリズム

ここではグラフ作成ツール全体のプログラム構成と図形要素を管理するためのデータ構造およびアルゴリズムについて解説する.

グラフ作成ツールのインターフェースは図 3 のように, 5 つの部分で構成されており, 処理プログラムは次の 5 つのイベント処理モジュールで構成されている.

- (1) 図形要素スク립ト挿入イベント このモジュールは, 図 3-(1) のプルダウンメニューに表示された図形要素サンプルリストから選択されたスク립トを記述エリアのカーソルポイントに挿入する処理を行う.
- (2) グラフ再描画イベント このモジュールは, グラフ作成ツールのメインプログラムであり, その流れ図を図 4 に示す. このイベントが発生すると, 図 3-(2) の図形要素スク립ト記述エリアに書かれたスク립ト文を解釈して, その中の数式部分

の構文解析によって JavaScript 形式の計算式に変換され、対応する図形要素データを生成し、図 3-(3) のグラフ表示キャンバスに関数グラフなどの図形を描画する仕組みになっている。その際に生成されたグラフのデータは、各図形要素をノードとするリスト構造をもち、保管される。図形要素ノードの共通するデータ構造を表 3 に示す。このノードのクラス名を MGnode と呼び、データ (変数) と処理プログラムルーチンをメンバーとして、各図形固有の情報を加えた継承クラスとして管理される。特に、メンバー変数 XYP は各図形要素をキャンバスに描画するメソッド put() が xy 平面の表示範囲 (定義域, 値域) を参照するための xy 平面領域へのポインターである。

表 3: 図形要素ノード (MGnode)

タイプ	メンバー名	内容
データ (変数)	name	図形要素名
	parent	前ノードへのポインタ (トップはルートノード)
	next	次ノードへのポインタ (末端は null)
	active	選択状態 (true: アクティブ, false: 非アクティブ)
	script	図形要素スクリプト
	XYP	xy 平面領域へのポインター (依存対象)
処理	put()	図形要素描画メソッド
	isOn(cx,cy)	指定座標 (cx,cy) との図形の重なり判定
	unActive()	選択 (アクティブ) 状態解除
	move(dx,dy)	図形の移動 (移動しない図形は未定義)

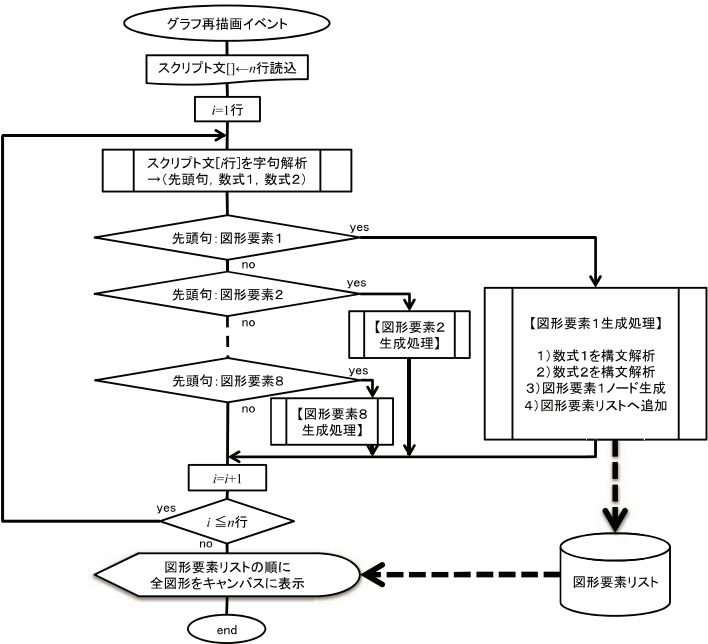


図 4: 図形再描画イベント処理

- (3) グラフ表示キャンバスイベント このモジュールは、グラフ表示キャンバス (図 3-(3)) 内の図形要素をマウスで選択または移動処理を行うためにある。
- (4) 表示オプション調整ボタンイベント このモジュールは、(3) で選択状態にある図形の表示オプション調整ボタン (図 3-(4)) によるイベントを処理する。
- (5) 数式入力インターフェース (MathTOUCH) 呼び出しイベント 図 3-(5) のボタン x^2 を受け付け、MathTOUCH を新規ウィンドウで起動し、ユーザが数式入力を完了した後、スクリプト記述エリアに数式を挿入する役目を担っている。

4.4 教育利用モデル事例

教育利用モデルとして、以下の 2 つの事例を紹介する。

4.4.1 グラフの平行移動の問題

本グラフ作成ツールを使うと、次のような問題が可能となる。

例題 1 : $y = x^2$ を x 軸方向に $+2$ だけ平行移動したグラフを描け。

普通は、 x 軸方向に a 、 y 軸方向に b だけ平行移動した場合の公式 $y - b = (x - a)^2$ から、 $y = (x - 2)^2$ を生徒に求めさせるだけの課題で終わることが多い。しかしこのツールを使えば、問題文にある「 $y = x^2$ 」を図 5(左) のようにすでに入力済みの状態で提示し、これに解答のグラフを描き加えさせることで、実際に導き出した式が正しく描画されるのかを簡単に確認させることができる。図 5(右) のスクリプトの 4 行目が生徒が入力したグラフである。

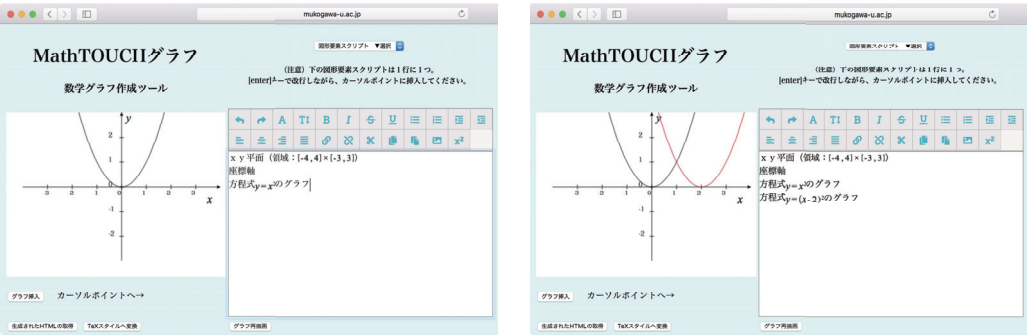


図 5: 例題 1 : グラフの平行移動の問題

4.4.2 グラフの接線の問題

例題 2 : 点 $(2, -1)$ を通る, $y = \frac{1}{3}x^2$ の接線およびその接点を描け.

これも普通は, 生徒に接線の式と接点座標を求めさせるだけで終わることが多いが, 本ツールを使えば, 問題文にある $y = \frac{1}{3}x^2$ のグラフと点 $(2, -1)$ を提示できる (図 6(左)) だけでなく, これに求めた解答の接線を描き加えさせ, 正しいことを確認させることができる (図 6(右)). 曲線 $y = f(x)$ 上の点 $(a, f(a))$ における接線の公式: $y - f(a) = f'(a)(x - a)$ から, $y = \frac{2}{3}a(x - a) + \frac{1}{3}a^2$ となり, これが点 $(2, -1)$ を通るので, 接線の一つの解 $y = \frac{1}{3}\{2(2 + \sqrt{7})x - 11 - 4\sqrt{7}\}$ および接点 $(2 + \sqrt{7}, \frac{11+4\sqrt{7}}{3})$ を生徒に求めさせるところまでは変わらない. しかし, 一般に従来のグラフ作成ツールのようなマウスでの位置調整で, 正確に指定することが困難な接点 $(2 + \sqrt{7}, \frac{11+4\sqrt{7}}{3})$ といった複雑な式の座標の点やグラフであっても, この試作ツールは正確な位置に描画することができるので, 正しさを確認できる. また図形要素スクリプト記述エリアで描いている描画命令文とグラフの内容が直結しているため, どの命令文がどの図形要素を描画させているのか, 描画されている図形要素が数学的にどのような式・座標・範囲なのかを理解しやすい.

以上の例題 1, 2 のように, このツールは計算を自動でしてくれないので, 生徒が自分で求めるという手順を奪わない. その一方で, 複雑な数値であっても (グラフィックス画面上) 正確な座標に描画できることから, 求めた答えが正しいのかを視覚的に確認できる.

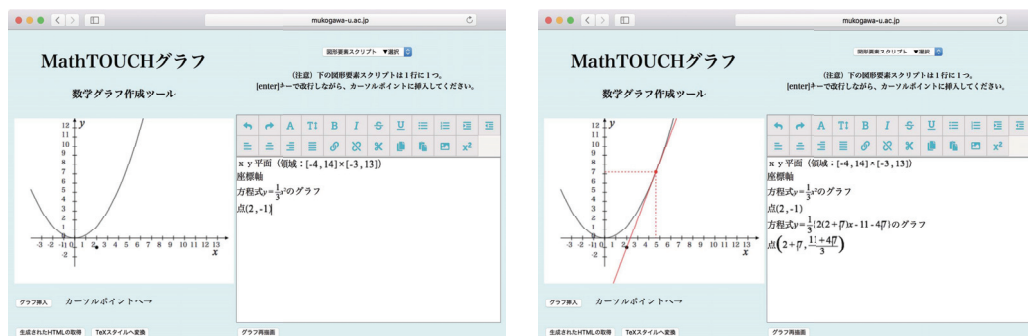


図 6: 例題 2 : グラフの接線の問題

5 まとめと今後の課題

今回開発したツールは, 自然な数学記述で 1 行ずつ描画したい図形要素を指定し, 記述された図形を描画するものであり, 現時点で一価関数までグラフ作成が可能である. また既存のツールと同様に, 図形要素の位置やサイズなどは描画後, インタラクティブに指定・調整できるようにしている.

既存のツールとの大きな違いは、数式入力インタフェース MathTOUCH の利点を活かし、自然な数式を使った入力・表現が可能になっていることである。そのため、比較的簡単に数式入力ができ、かつ図形描画を指示するスクリプトが自然な数学表現に近く、読みやすい。

これによって、数学初学者・学習者である生徒・学生が意味を考えながら、作図しやすくなるため、数学ノートや答案として記録するために使えるものと期待する。さらに、図のように、教科書のような自然な数学記述によってグラフ作成ができることから、教育指導現場のツール (プリント作成, 授業内での説明に利用など) としても有効ではないかと考える。

今後は不足している機能の充実と、教育実践による有効性の検証を行いたい。

参考文献

- [1] 文部科学省公式サイト:「3. デジタル教科書の制度化について」,
http://www.mext.go.jp/a_menu/shotou/kyoukasho/gaiyou/04060901/1349317.htm
(2018.10.11 参照).
- [2] 福井哲夫:「数式のインテリジェントな線形入力方式」, 京都大学数理解析研究所講
究録, Vol. 1780, pp.160–171, 2012.
- [3] 福井哲夫:「数式のインテリジェントな線形入力方式と評価」, 日本数式処理学会
数式処理, Vol. 18 No.2, pp.47–50, 2012.
- [4] Apple:「Grapher ユーザガイド」,
<https://support.apple.com/jajp/guide/grapher/welcome/mac> (2018.10.11 参照).
- [5] GeoGebra:「GeoGebra 数学アプリ」, <http://www.geogebra.org/> (2018.10.11 参照).
- [6] WOLFRAM:「WOLFRAM MATHEMATICA」,
<https://www.wolfram.com/mathematica/> (2018.10.15 参照).
- [7] W3C:「HTML 5.2」, <https://www.w3.org/TR/index.html> (2018.10.15 参照).
- [8] 白井詩沙香, 仲村裕子, 福井哲夫:「数式自動採点システムにおける数式入力イン
タフェースの提案と評価」, 情報処理学会論文誌 教育とコンピュータ (TCE), Vol.
1, No.3, pp.11–21, 2015.
- [9] 白井詩沙香, 福井哲夫:「数式自動採点システム STACK における数式入力方法の
改善」, コンピュータ&エデュケーション Vol. 37, pp.85–90, 2014.
- [10] Fukui, T., Shirai, S.:「Predictive Algorithm for Converting Linear Strings to Gen-
eral Mathematical Formulae」, Human Interface and the Management of Informa-
tion: Supporting Learning, Decision-Making and Collaboration, Lecture Notes in
Computer Science, Vol. 10274, Springer, pp.15–28, 2017.